CLAIMS

1. A method of testing compliance of a device with a bus protocol, the method comprising the steps of:

(a) reading a configuration file containing predetermined parameters identifying the type of the device and capabilities of the device;

(b) employing a configuration engine to dynamically generate a test environment for the device by creating selected test components which are coupled via the bus with a representation of the device to form the test environment, the test components being selected dependent on the configuration file;

(c) causing a test sequence to be executed; and

(d) monitoring signals passed between the representation of the device and one or more of the test components during execution of the test sequence to generate result data indicating compliance with the bus protocol.

2. A method as claimed in Claim 1, wherein the configuration file is selected from a set of configuration file templates, the set containing a configuration file template for each type of device, and each configuration file having a number of entries for providing configuration information specific to the device.

3. A method as claimed in Claim 1, wherein said step (d) comprises the step of employing a protocol checking component to check that signals passed between the representation of the device and one or more of the test components during execution of the test sequence do not violate a set of predetermined rules of the bus protocol.

4. A method as claimed in Claim 1, wherein said step (d) comprises the step of employing a coverage monitoring component to monitor signals passed between the representation of the device and one or more of the test components during execution of the test sequence to determine whether a set of coverage points are observed.

5. A method as claimed in Claim 4, wherein the set of coverage points is configured dependent on the configuration file read at said step (a).

6. A method as claimed in Claim 5 wherein said step (d) comprises the step of employing a protocol checking component to check that signals passed between the representation of the device and one or more of the test components during execution of the test sequence do not violate a set of predetermined rules of the bus protocol, and wherein, if all coverage points in the set have been observed without violating any of the set of predetermined rules of the bus protocol, the method further comprises the step of generating an output confirming compliance with the bus protocol.

7. A method as claimed in Claim 1, wherein at said step (b) the step of creating selected test components comprises selecting the test components to be created in dependence on the type of device to be tested.

8. A method as claimed in Claim 7, wherein at least one of the test components has associated therewith a plurality of behaviours that it may exhibit, the choice of behaviour being determined when that test component is created dependent on the type of device to be tested.

9. A method as claimed in Claim 1, wherein the test sequence is a user-definable test sequence developed for the device to be tested.

10. A method as claimed in Claim 1, wherein the representation of the device is created within an interface module, and said step (b) of generating the test environment includes mapping signals within the interface module to signals within the test environment, the mapping being defined within the configuration file.

11. A method as claimed in Claim 10, wherein the configuration file identifies a level of hierarchy of the representation of the device within the interface module to facilitate the mapping of signals.

12. A method as claimed in Claim 1, further comprising the step of:

providing a trickbox component compatible with the bus protocol and provided with a general-purpose input/output interface.

13. A method as claimed in Claim 1, wherein the type of device that may be tested comprises a master, a slave, an arbiter or a decoder.

14. A method as claimed in Claim 1, wherein the bus protocol is the ARM AMBA bus protocol, the bus comprises a system bus and a peripheral bus, and the type of device which may be tested comprises a system bus master, a system bus slave, a peripheral bus master, a peripheral bus slave, an arbiter or a decoder.

15. A method as claimed in Claim 1, wherein the representation of the device is a Register Transfer Language (RTL) representation.

16. A computer program operable to configure a processing unit to perform a method of testing compliance of a device as claimed in Claim 1.

17. A carrier medium comprising a computer program as claimed in Claim 16.

18. A data processing system for testing compliance of a device with a bus protocol, the system comprising:

memory for storing a configuration file containing predetermined parameters identifying the type of the device and capabilities of the device; and

a processing unit arranged to perform the steps of:

(i) dynamically generating a test environment for the device by creating selected test components which are coupled via the bus with a representation of the device to form the test environment, the test components being selected dependent on the configuration file;

(ii) executing a test sequence; and

(iii) monitoring signals passed between the representation of the device and one or more of the test components during execution of the test sequence to generate result data indicating compliance with the bus protocol.